EXTRACTION OF INTERESTING ASSOCIATION RULES USING GENETIC ALGORITHMS

Peter P. Wakabi-Waiswa*
Faculty of Computing and IT, Makerere University

Venansius Baryamureeba,
Faculty of Computing and IT, Makerere University.

The process of discovering interesting and unexpected rules from large data sets is known as association rule mining. The typical approach is to make strong simplifying assumptions about the form of the rules, and limit the measure of rule quality to simple properties such as support or confidence. Support and confidence limit the level of interestingness of the generated rules. Comprehensibility, J-Measure and predictive accuracy are metrics that can be used together to find interesting association rules. Because these measures have to be used differently as measures of the quality of the rule, they can be considered as different objectives of the association rule mining problem. The association rule mining problem, therefore, can be modelled as a multi-objective problem rather than as a single-objective problem. In this paper we present a Pareto-based multi-objective evolutionary algorithm rule mining method based on genetic algorithms. Predictive accuracy, comprehensibility and interestingness are used as different objectives of the association rule mining problem. Specific mechanisms for mutations and crossover operators together with elitism have been designed to extract interesting rules from a transaction database. Empirical results of experiments carried out indicate high predictive accuracy of the rules generated..

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications --- Data Mining; F.2.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity --- Non-numerical Algorithms and Problems --- *Sorting and searching*; G.4 [Mathematics of Computing]: Mathematical Software --- *Algorithm design and analysis;* G.3 [Mathematics of Computing]: Probability and Statistics --- *Probabilistic algorithms* I.2.8 [Artificial Intelligence]:Problem Solving, Control Methods, and Search - *Heuristic methods*; J.1 [Computer Applications]: Administrative Data Processing-- Business, education, marketing;
General Terms: Association Rule Mining,
Additional Key Words and Phrases: Interestingness, Multi-Objective Evolutionary Algorithms, Genetic Algorithms, Comprehensibility, interestingness and surprise.

IJCIR Reference Format: Peter P. Wakabi-Waiswa and Venansius Baryamureeba. Extraction of Interesting Association Rules Using Genetic Algorithms. International Journal of Computing and ICT Research, Vol. 2, No. 1, pp. 26 – 33. http:www.ijcir.org/volume2-number1/article4.pdf..

---

1. INTRODUCTION

Association rule mining (ARM) is one of the core data mining techniques. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. The major aim of ARM is to find the set of all subsets of items or attributes that frequently occur in many database records or transactions, and additionally, to extract rules on how a subset of items influences the presence of another subset. ARM

* Author's Address: Peter P. Wakabi, Department of Computer Science, Faculty of Computing and IT, Makerere University, P.O. Box 7062, Kampala, Uganda, pwakabi@gmail.com
Venansius Baryamureeba, Department of Computer Science, Faculty of Computing and IT, Makerere University, P.O. Box 7062, Kampala, Uganda, barya@cit.mak.ac.ug , www.cit.ac.ug

algorithms discover high-level prediction rules in the form: IF the condition of the values of the predicting attributes are true, THEN predict values for some goal attributes.

The task of mining association rules over market basket data was first introduced by Agrawal et al. [1993], can be formally stated as follows: Let I ={ i1 , i2, ..., im} be the set of database items and T={ t1, t2, ..., tm} be the set of transactions in the database, D, with each transaction ti having a unique identifier and containing a set of items, called an itemset. An association rule is a conditional implication among itemsets, X→Y, where X and Y are itemsets and X ∩ Y = □. An itemset can be a single item (e.g. sugar) or a set of items (e.g. runners, shorts, sugar and mineral water). An itemset with k items is called a k-itemset. A subset of k elements is called a k-subset. An itemset is said to be frequent or large if its support is more than a user specified minimum support value. The support of an itemset is the percentage of transactions in D that contain the itemset. The confidence of an association rule, given as support (X ⋃ Y)/support (X), is the conditional probability that a transaction contains Y given that it also contains X.

In the following example of a bookstore sales database, the association rule mining task is exemplified. There are five different items (authors of novels that the bookstore deals in), I = {A, C, D, T, W}. There are six customers in the database who purchased books by these authors. The table below shows all frequent itemsets containing at least three authors (i.e. minimum − support = 50 %.). It also shows the set of all transactions

Item	Abbreviation
John Ayo	A
Alfred Chihoma	C
Bernard Dungu	D
Thomas Babatunde	T
Peter Walwasa	W

Fig. 1: Items

Transaction	Items
1	ACTW
2	CDW
3	ACTW
4	ACDW
5	ACDTW
6	CDT

Fig. 2: Database Transactions

Support	Itemsets
100%	C
83%	W, CW
67%	A,D,T,AC,A, CD, CT,ACW
50%	AT,DW,TW,ACT,ATW CDW,CTW,ACTW

Fig. 3: Frequent Itemsets

A → C(4/4)	AC → W(4/4)	TW → C(3/3)
A → W(4/4)	AT → C(3/3)	AT→ CW(3/3)
A → CW(4/4)	AT → W(3/3)	TW→ AC(3/3)
D → C(4/4)	AW → C(4/4)	ACT→ W(3/3)
T → C(4/4)	DW → C(3/4)	ATW→ C(3/3)
W → C(5/5)	TW → A(3/3)	CTW→ A(3/3)

Fig. 4: Association Rules

Considering the first association rule A → C [support =50%, confidence=67%], which says that 50% of people buy books authored by John Ayo (A) and those by Alfred Chihoma (C) together, and 67% of the people who buy books written by John Ayo (A) also purchase those by Alfred Chihoma (C).

According to Zaki [1999] the mining task involves generating all association rules in the database that have a support greater than the minimum support (the rules are frequent) and have a confidence greater than minimum confidence (rules are strong). The ARM problem is an NP-Hard problem because finding all frequent itemsets (FI's) having a minimum support results in a search space of 2m , which is exponential in m, where m is the number of itemsets. The final step involves generating strong rules having a minimum confidence from the frequent itemsets. It also includes generating and testing the confidence of all rules. Since each subset of X as the consequent must be considered, the rule generation step's complexity is O(r.2ℓ), where r is the number of frequent itemsets, and ℓ is the longest frequent itemset.

Traditionally, ARM was predominantly used in market-basket analysis but it is now widely used in other application domains including customer segmentation, catalogue design, store layout, and telecommunication alarm prediction. ARM is computationally and I/O intensive. The

number of rules grows exponentially with the number of items. Because data is increasing in terms of both the dimensions (number of items) and size (number of transactions), one of the main attributes needed in an ARM algorithm is scalability: the ability to handle massive data stores. Sequential algorithms cannot provide scalability, in terms of the data dimension, size, or runtime performance, for such large databases.

In this paper, we shall deal with the ARM problem as a multi-objective problem rather than as a single one and try to solve it using multi-objective evolutionary algorithms (MOEA) with emphasis on genetic algorithms (GA). The main motivation for using GAs is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often used in data mining tasks. Multi-objective optimisation with evolutionary algorithms is well discussed by Fonseca and Fleming [1998] and Freitas [2003].

Throughout this paper, we use the following notation. The support of an itemset X, which is the number of transactions in which that itemset occurs as a subset, is denoted by $\sigma(X)$. The rule's support is the joint probability of a transaction containing both X and Y, and is denoted by $\sigma(X \cup Y)$. The confidence of the rule (also known as the predictive accuracy of a rule) is the conditional probability that a transaction contains Y, given that it contains X and is given as $\sigma(X \cup Y)/\sigma(X)$. !denotes a logical negation.

The rest of this paper is organised as follows. In Section 2 we provide an overview of work related to the association rule mining problem. In Section 3 we discuss the proposed MOEA. In Section 4, the proposed algorithm is presented. In Section 5 the analysis of the results is presented. Section 6 is the conclusion.

2. Related Work

The Association Rule mining problem was introduced in 1993 by Agrawal et al. [1993]. Agrawal et al. [1993] developed the Apriori algorithm for solving the association rule mining problem. Most of the existing association rule mining algorithms are improvements to the Apriori algorithm [Ghosh and Nath 2004; Zhao and Bhowmick 2003], and are referred to as Apriori-based algorithms. These algorithms work on a binary database, termed as the market basket database. On preparing the market basket database, every record of the original database is represented as a binary record where the fields are defined by a unique value of each attribute in the original database. The fields of this binary database are often termed as an item. For a database having a huge number of attributes and each attribute containing a lot of distinct values, the total number of items will be huge. Storage requirements resulting from the binary database is enormous and as such it is considered one of the limitations of the existing algorithms.

The Apriori-based algorithms work in two phases. The first phase is for frequent itemset generation. The itemsets are used for generating interesting rules. A rule is said to be interesting if its confidence is above a user's specified minimum confidence. Frequent itemsets are generated by searching from all-possible itemsets the itemsets whose support is greater than the user specified minimum support. If the value of minimum support is too high, the number of frequent itemsets generated will be less, and thereby resulting in generation of few rules. And, if the value is too small, then almost all possible itemsets will become frequent and thus a huge number of rules may be generated. This causes inference basing on these rules to be difficult. After detecting the frequent itemsets in the first phase, the second phase generates the rules using minimum confidence. Confidence factor or predictive accuracy of a rule is defined as:

$$\text{Confidence} = \sigma(X \cup Y)/\sigma(X) \quad (1)$$

Another limitation of the Apriori-based algorithms is the encoding scheme where separate symbols are used for each possible value of an attribute [Ghosh and Nath 2004]. This encoding scheme may be suitable for encoding the categorical valued attributes, but not for encoding the numerical valued attributes as they may have different values in every record. To avoid this situation, some ranges of values may be defined. For each range of values an item is defined. This approach is also not suitable for all situations. Defining the ranges will create yet another problem, as the range of different attributes may be different.

Existing Apriori-based algorithms, measure the quality of a generated rule by considering only one evaluation criterion, i.e., confidence factor or predictive accuracy [Ghosh and Nath 2004]. This criterion evaluates the rule depending on the number of occurrences of the rule in the entire database.

In this work we propose to develop an algorithm that uses comprehensibility, interestingness, and predictive accuracy as measures of the quality of the rules and apply them as objectives to model the association rule mining problem as a multi-objective problem. The details of these measures are given in the following section.

3. Multi-objective optimization and rule mining problems

It is not an easy task to find a single solution for a multi-objective problem. In such situations the best approach is to find a set of solutions depending on non-dominance criterion. At the time of taking a decision, the solution that seems to fit better, depending on the circumstances can be chosen from the set of these candidate solutions. A solution, say a, is said to be dominated by another solution, say b, if and only if the solution b is better or equal with respect to all the corresponding objectives of solution a, and b is strictly better in at least one objective. Here solution b is called a non-dominated solution. So it will be helpful for the decision-maker, if a set of such non-dominated solutions can be found. Vilfredo Pareto suggested this approach for solving the multi objective problem. Optimization techniques based on this approach are called Pareto optimization techniques. Based on this idea, several genetic algorithms were designed to solve general multi-objective problems [Ghosh and Nath 2004].

In association rule mining, if the number of conditions involved in the antecedent part is less than the one in the consequent part, the rule is more comprehensible. We therefore require an expression where the number of attributes involved in both parts of the rule has some effect. The following expression can be used to quantify the comprehensibility of an association rule

$$\text{Comprehensibility} = \log(1+|Y|)+\log(1+|X \cup Y|) \quad (2)$$

Here, $|Y|$ and $|X \cup Y|$ are the number of attributes involved in the consequent part and the total rule, respectively.

It is important that we extract only those rules that have a comparatively less occurrence in the entire database. Such a surprising rule may be more interesting to the users; which again is difficult to quantify. According to Liu et al. [2000], the interestingness issue has long been identified as an important problem in data mining. It refers to finding rules that are interesting/useful to the user, not just any possible rule. The reason for its importance is that, in practice, it is all too easy for a data mining algorithm to discover a huge range of rules most of which are of no interest to the user.

To find interestingness, the data set is divided based on each attribute present in the consequent part. Since a number of attributes can appear in the consequent part and they are not predefined, this approach may not be feasible for association rule mining. So a new expression is defined which uses only the support count of the antecedent and the consequent parts of the rules, and is defined as

$$I = [\sigma(X \cup Y)/\sigma(X)] \times [\sigma(|X \cup Y|)/\sigma(Y)] \times [1 - \sigma(X \cup Y)/|D|)] \quad (3)$$

where I is interestingness and $|D|$ is the total number of records in the database, $\sigma(X \cup Y)/\sigma(X)$ gives the probability of generating the rule depending on the antecedent part, $\sigma(|X \cup Y|)/\sigma(Y)$ gives the probability of generating the rule depending on the consequent part, and $\sigma(X \cup Y)/|D|)$ gives the probability of generating the rule depending on the whole dataset. This means that the complement of this probability will be the probability of not generating the rule. Thus, a rule having a very high support count will be measured as less interesting.

On top of comprehensibility, interestingness and support count there are other metrics that can be used in generating more informative rules. The other good metrics include J-Measure and entropy. The J-Measure is a good indicator of the information content of the generated rules. In rule inference we are interested in the distribution of the rule "implication" variable Y, and especially its two events y and complement $\bar{y}$. The purpose is to measure the difference between the priori distribution f(y), i.e. $f(Y = y)$ and $f(Y \neq y)$, and the posteriori distribution $f(Y \mid X)$. The J-Measure gives the average mutual information between the events y and $f(Y = y)$. The J-Measure shows how dissimilar our a priori and posteriori beliefs are about Y meaning that useful rules imply a high degree of dissimilarity.

The J-Measure is calculated as:

$$JM = f(y|x).\log 2\left(\frac{f(y|x)}{f(y)}\right) + (1 - f(y|x).\log 2\left(\frac{(1 - f(y|x))}{1 - f(y)}\right) \quad (4)$$

where JM is the J-Measure.

The entropy, on the other hand, measures the level of surprise in the rule(s). Entropy, also called surprisal, measures the amount of randomness or surprise or uncertainty. It is calculated as follows: Given probabilities p1, p2, ..., pn whose sum is 1:

$$\text{Entropy H(pi)} = \sum_{i=0}^{n} p(i)\log 2.p(i) \quad (5)$$

4 Genetic Algorithms with Modifications

We propose to solve the association rule-mining problem with a Pareto based multiple-objective genetic algorithm. The possible rules are represented as chromosomes and a suitable encoding/decoding scheme has been defined. Genetic algorithms (GAs) for rule discovery can be divided into two broad approaches, the Michigan approach and the Pittsburgh approach [Dehuri et al. 2006]. The biggest distinguishing feature between the two is that in the Michigan approach (also referred to as Learning Classifier Systems) an individual is a single rule, whereas in the Pittsburgh approach each individual represents an entire set of rules.

In the context of this research the use of the term Michigan approach will denote any approach where each GA individual encodes a single prediction rule. The choice between these two approaches strongly depends on which kind of rule is to be discovered. This is related to which kind of data mining task being addressed. Suppose the task is classification. Then evaluate the quality of the rule set as a whole, rather than the quality of a single rule. In other words, the interaction among the rules is important. In this case, the Pittsburgh approach seems more natural [Frietas 2002].

On the other hand, the Michigan approach might be more natural in other kinds of data mining tasks. An example is a task where the goal is to find a small set of high-quality prediction rules, and each rule is often evaluated independently of other rules. The Pittsburgh approach directly takes into account rule interaction when computing the fitness function of an individual. However, this approach leads to syntactically-longer individuals, which tends to make fitness computation more computationally expensive. In addition, it may require some modifications to standard genetic operators to cope with relatively complex individuals.

By contrast, in the Michigan approach the individuals are simpler and syntactically shorter. This tends to reduce the time taken to compute the fitness function and to simplify the design of genetic operators. However, this advantage comes with a cost. First of all, since the fitness function evaluates the quality of each rule separately, now it is not easy to compute the quality of the rule set as a whole - i.e. taking rule interactions into account. Another problem is that, since we want to discover a set of rules, rather than a single rule, we cannot allow the GA population to converge to a single individual which is what usually happens in standard GAs. This introduces the need for some kind of niching method. The need for niching in the Michigan approach may be avoided by running the GA several times, each time discovering a different rule. The drawback of this approach is that it tends to be computationally expensive.

We have, therefore, used a modified Michigan encoding/decoding scheme which associates two bits to each attribute. If these two bits are 00 then the attribute next to these two bits appears in the antecedent part and if it is 11 then the attribute appears in the consequent part. And the other two combinations, 01 and 10 will indicate the absence of the attribute in either of these parts. So the rule ACF → BE will look like 00A 11B 00C 01D 11E 00F. In this way we can handle variable length rules with more storage efficiency, adding only an overhead of 2k bits, where k is the number of attributes in the database. The decoding is performed as follows:

$$DV = \text{minval} + (\text{maxval} - \text{minval}) \times (\sum (2i-1 \times \text{ith bitvalue})/(2n - 1)) \quad (6)$$

where DV is the decoded value; $1 \leq i \leq n$ and n is the number of bits used for encoding; minval and maxval are minimum and maximum values of the attribute; and bitvalue is the value of the bit in position i. For brevity, this encoding scheme will not deal with relational operators and as such

the rules generated from this formula will not include relational operators.

Due to the fact that there may be a large number of attributes in the database, we propose to use multi-point crossover operator. There are some difficulties to use the standard multi-objective GAs for association rule mining problems. In case of rule mining problems, we need to store a set of better rules found from the database. Applying the standard genetic operations only, the final population may not contain some rules that are better and were generated at some intermediate generations. The better rules generated at intermediate stages should be kept. For this task, an external population is used. In this population no genetic operation is performed. It will simply contain only the non-dominated chromosomes of the previous generation. At the end of first generation, it will contain the non-dominated chromosomes of the first generation. After the next generation, it will contain those chromosomes, which are non-dominated among the current population as well as among the non-dominated solutions till the previous generation.

The scheme applied here for encoding/decoding the rules to/from binary chromosomes is that the different values of the attributes are encoded and the attribute names are not. For encoding a categorical valued attribute, the market basket encoding scheme is used. For a real valued attribute their binary representation can be used as the encoded value. The range of values of that attribute will control the number of bits used for it.

The archive size is fixed, i.e., whenever the number of non-dominated individuals is less than the predefined archive size, the archive is filled up by dominated individuals. Additionally, the clustering technique used does not loose boundary points.

4.1 Fitness Assignment

As discussed earlier, we use a set of three complementary metrics as criteria for filtering out interesting rules. We combine these metrics into an objective fitness function. The complementary set of measures include confidence defined in equation (1), comprehensibility defined in equation (2) and J-Measure defined in equation (4). The fitness function is calculated as the arithmetic weighted average confidence, comprehensibility and J-Measure. The fitness function (f(x)) is given by:

$$f(x) = \frac{W_1 * Comprehensibility + W_2 * (J - Measure) + W_3 * Confidence}{W_1 + W_2 + W_3} \quad (7)$$

where $W_1, W_2 W_3$ are user-defined weights.

4.2 Environmental Selection

Due to the fact that genetic algorithms (GA) are rooted in natural genetics, most of the terminologies in the GA field are analogous to that used in natural evolution. In a genetic algorithm, the environment is the problem under consideration and each of the organisms is a solution to the problem. Each potential solution to a GA problem is called an individual. Each individual is made up of genes (often represented as a set of numbers).

Two things must be determined in order to apply a genetic algorithm to a given problem: i) a genetic code representation and ii) a fitness or objective function, which assigns a quality measure to each solution according to its performance. The encoding of the parameters in genetic algorithms depends on the problems at hand.

A group of individuals, called a population, is stored and modified with each iteration of the algorithm. In GA's iterations are referred to as generations. The selection of these individuals is based on their fitness. Individuals in each new generation carry forward genes from the previous generations, and the individuals which are more fit will tend to survive and reproduce.

The number of individuals contained in the archive is constant over time, and the truncation method prevents boundary solutions being removed. During environmental selection, the first step is to copy all non-dominated individuals, i.e., those which have a fitness lower than one, from archive and population to the archive of the next generation. If the non-dominated front fits exactly into the archive the environmental selection step is complete. In case the archive is too small, the best dominated individuals in the previous generation and population are copied to the new archive. Otherwise, truncate the archive.

5 Experiments

Experiments were conducted using real-world Zoo dataset*. For brevity, the data used is of a categorical nature. The Zoo database contains 101 instances corresponding to animals and 18 attributes. The attribute corresponding to the name of the animal was not considered in the evaluation of the algorithm. This was mainly due to its descriptive nature. The attribute from the datasets that were used for analysis include: hair [H], feathers [F], eggs [E], milk [M], predator [P], toothed [TH], domestic [D], backbone [B], fins [N], legs [L], tail [T], cat-size [C], airborne [A], aquatic [Q], breathes [BR], venomous [V], and type [Y].

Default values of the parameters are: Population size = 40, Mutation rate = 0.5, Crossover rate =0.8, Selection in Pareto Archive (elitism) = 0.5. The stopping criterion used is the non evolution of the archive during 10 generations, once the minimal number of generations has been over passed.

5.1 Results and Discussion

In the following table are the results of the experiments conducted. In the first column is the discovered rule, in the second column is the rule's comprehensibility, in the third is the J-Measure of the rule and in the last column is the predicative accuracy of the rule. In these tests, different predictions were made by combining different attributes to determine a result.

The following are results from the given data

Discovered Rule	Compre-hensibility	J-Measure	Predictive Accuracy
If (!H and E and !M and B and T and D) Then (!P)	0.97	0.84	0.90
If(!A and Q and B and C) Then (P)	0.95	0.77	0.94
If(E and A and P and !V) Then (!D)	0.96	0.66	0.98
If(!E and !Q and !T) Then (D)	0.97	0.93	0.50
If(!E and !V and !D) Then (Y=1)	0.95	0.87	0.98
If(F and !V and !D) Then (Y=2)	0.94	0.93	0.97
If(E and !Q and P and TH and !N and !D and !C) Then(Y=3)	0.94	0.97	0.83
If(Q and !BR and !V and T) Then(Y=4)	0.93	0.93	0.95
If(!A and Q and T and BR and !C) Then(Y=5)	0.94	0.98	0.80
If(A=1)and(!N)and(!T) Then(Y=6)	0.93	0.96	0.90
If(P)and(BR)and(!T)and (!D) Then(Y=7)	0.95	0.95	0.92

As it is indicated in the results table, overall the discovered rules have a high predictive accuracy and are quite interesting. Four rules have a predictive accuracy of over 90% while seven rules have a J-Measure of over 90%. Three rules have a high predictive accuracy of over 80%. Only two rules have a predictive accuracy of less than 50%.

6 Conclusion and Future Work

We have dealt with a challenging NP-Hard association rule mining problem of finding interesting association rules. The results reported in this paper are very promising since the discovered rules are of a high comprehensibility, high predictive accuracy and of a high interestingness values. However, a more extensive empirical evaluation of the proposed multi-objective algorithm will be the objective of our future research. We also intend to extend the algorithm proposed in this paper to cope with continuous data since the current one handles only categorical data. The incorporation of other interestingness measures mentioned in the literature is also part of our planned future work.

References

AGRAWAL, R., IMIELINSKI, T., AND SWAMI, A. 1993. Mining Association Rules Between Sets of Items in Large Databases. Proc. of the 1993 ACM SIGMOD Conf. on Management of Data.

AGRAWAL, R., SRIKANT, R.1994. Fast Algorithms for Mining Association Rules. Proc. of the 20th Int'l Conf. on Very Large Data Bases.

* UCI Machine Learning Repository; http://www.cs.uci.edu/mlearn/MLRepository.html, accessed June 2006

LIU, B., HSU, W., CHEN, S., AND MA, Y. 2000. Analyzing the Subjective Interestingness of Association Rules. IEEE Intelligent Systems.

FONSECA, M. C., FLEMING J. P. 1998. Multi-objective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part I: A Unified Formulation. IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, 28(1):26-37

GHOSH, A., NATH, B. 2004. Multi-objective rule mining using genetic algorithms. Information Sciences 163 pp 123-133

FREITAS, A. A. 2003. A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. Advances in evolutionary computing: theory and applications, Pp 819 - 845

DEHURI, S., JAGADEV, A. K., GHOSH A. AND MALL R. 2006. Multi-objective Genetic Algorithm for Association Rule Mining Using a Homogeneous Dedicated Cluster of Workstations. American Journal of Applied Sciences 3 (11): 2086-2095, 2006 ISSN 1546-9239

KHABZAOUI, M., DHAENES, C., AND TALBI, E. 2005. Parallel Genetic Algorithms for Multi-Objective rule mining. MIC2005. The 6th Meta-heuristics International Conference, Vienna, Austria.

LIU, B., HSU, W., CHEN, S. AND MA, Y. 2000. Analyzing the Subjective Interestingness of Association Rules. IEEE Intelligent Systems.

ZAKI, M.J. 2001. Generating non-redundant association rules. In Proceedings of the Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY: ACM, 34-43.

ZHAO, Q., BHOWMICK, S. S. 2003. Association Rule Mining: A Survey. Technical Report, CAIS, Nanyang Technological University, Singapore, No. 2003116 , 2003.

ZITZLER, E., DEB, K., AND THIELE,L. 1998. An evolutionary Algorithm for Multi-objective Optimization: The Strength Pareto Approach

ZITZLER, E., DEB, K., AND THIELE, L. 2000. Comparison of Multi-objective Evolutionary Algorithms: Empirical Results. Evolutionary Computation Vol. 8 Number 2 pp 173 -195

ZITZLER, E., LUAMANNS, M., AND THIELE, L. 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Computer Engineering and Networks Laboratory (TIK), TIK-Report 103